Machine Learning Project Report

The Income Prediction Problem

Submitted By:
Akshay Tambe (apt321@nyu.edu)
Snahil Singh (ss1181@nyu.edu)

NYU TANDON SCHOOL OF ENGINEERING

Problem Statement

We're given data from the United States Census and our goal is to predict the yearly income from wages (in dollars) of a person in the United States based on other features of the person. It's a regression problem scenario where we need to predict the wages of a person in the testing dataset using the hypothesis developed from the training dataset.

Data Pre-processing

Data Cleaning

There are 17 features in the dataset and most of them contain missing values marked as "?", which need to be handled.

From the exploratory analysis, we handled the features in the following manner:

No Null values:

There are columns like *idnum, age, interestincome, marital, sex, ancestry, wages* with no N/A or null values.

Handling N/A Values:

1. ***workerclass*:**
    a. There are 22 people with *age* less than 16 yrs old and *workerclass* equal to N/A, so we assign a new category, *workerclass* = *'10'* to them. Also from the dataset given, people with *age* < 16 yrs, have *wages* = 0.
    b. We can see from the dataset that the columns *workerclass, traveltimetowork, vehicleoccupancy, meansoftransport, workarrivaltime, hoursworkperweek, industryworkedin,* all contain some N/A values and for each one of them it says that they are either not working or a subset of them is for people who are currently not working or have worked more than five years ago, so we assign a new category *workerclass* = *'11*' to people where all these column values are N/A together, thus separating the non-working class.
2. ***traveltimetowork*:**
    It is N/A for people who have never worked or who work from home, so that implies they don't travel and hence their *traveltimetowork* can be considered 0.
3. ***vehicleoccupancy*:**
    a. A new category '0' can be added for people who don't travel, i.e. their *meansoftransport* is either walk (Category = '10') or worked at home (Category = '11') and also for people who don't work, which we derive from *workerclass* with Category = '10' and '11'.
    b. People using *meansoftransport* as Category = '2' (Bus or trolley bus)', '4' subway or elevated, '5' (Railroad) are all public means of transport and so we assign the Category = '10' (in 10 person or more carpool).
    c. A new category ='10', added for the rest of the rows where *vehicleoccupancy* is N/A.
4. ***meansoftransport*:**
    A new category '0' added denoting means of transport for people not working currently.
5. ***schoolenrollment***:
    It is N/A for children with age less than 3 yrs old, generally, minimum age 5 is required to enroll in a school, so we assign a new category '0' (No, less than 3 yrs old) to them.
6. ***educationalattain*:**
    It is N/A for people with age less than 3 yrs old, so we assign a new category '0' (No education attained, less than 3 yrs old) to them.
7. ***workarrivaltime*:**
    It contains N/A for people who don't work or who work from home, as these people don't need to travel to work, we assign a new category '0' to them.

8. ***hoursworkperweek*:**
   It is N/A for people who don't work or didn't work in past 12 months or are less than 16 yrs old, so we can consider them not working currently, and assign a new category '0' (non-worker) to them.
9. ***degreefield*:**
   People with degree less than Bachelor's degree are assigned a new category '1000'.
10. ***industryworkedin*:**
    Here it has N/A value when a person doesn't work or last worked more than 5 yrs ago or less than 16 yrs old or not in labor force, so it can be assigned a new category = '0100'.

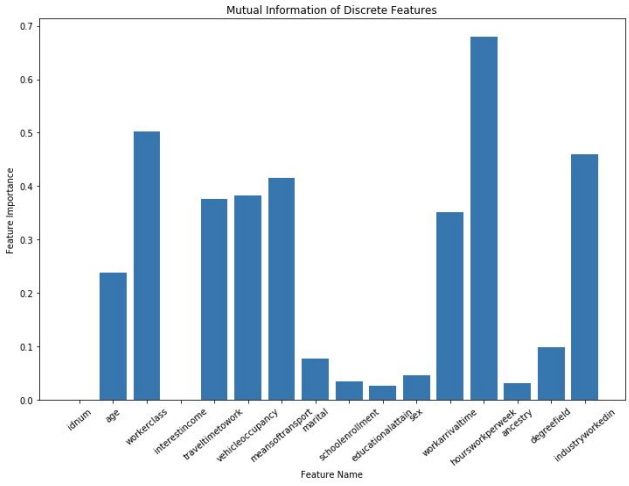
Separating Categorical and Numerical features:

1. Categorical features: *workerclass, vehicleoccupancy, meansoftransport, marital, schoolenrollment, educationalattain, sex, workarrivaltime, ancestry, degreefield, industryworkedin.*

2. **Numerical features:** *idnum, age, interestincome, traveltimetowork, hoursworkperweek, wages.*

Feature Engineering

1. **Features dropped based on Mutual Information:**



Sci-kit learn's *mutual_info_regression()*[1] function was used to get the mutual information of each feature with the target variable

Based on the information gain plot above, we dropped the following columns - *idnum, interestincome, ancestry, schoolenrollment.*

2. **Feature Extraction**
   This step allows us to capture important information in a dataset much more effectively than original dataset.
   a. **Productivity:**
      A new attribute *'Productivity'* has been added to the dataset, it's derived from the following attributes - age and education attained. A person's income depends upon his capacity to work and how educated he is. The age of a person tells us about the capacity of that person and is inversely proportional to his productivity. Education attained is directly proportional to the productivity of a person and it increases with a person's income. We take the class of the 'educationalattain' as the weight assigned to that education degree, so higher the class, higher is the weight of that degree. Then we use this along with the age of a person to compute a person's productivity level, i.e.

      Productivity = k (Education attained / Age) *(k being any proportionality constant)*

We can see what impact do age and education attained have on a person's income below:

b. WageClass:

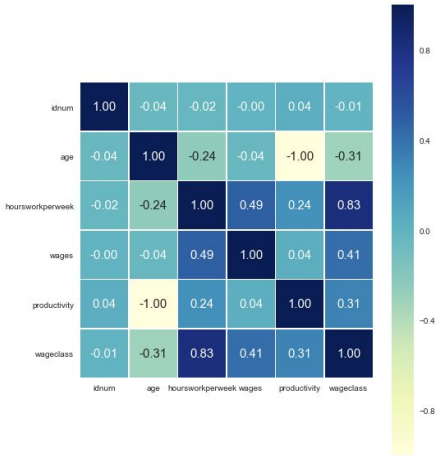
A binary classifier is used to classify if a person earns or not, '0' for a person who does not earn and '1' otherwise. If a person is classified to be earning, then we use regression to predict his income else we mark his income as zero.

These two attributes 'Productivity' and 'wageclass' are added as numerical attributes to the feature set and used further to develop the model.

Data Transformation

Scaling

We standardize the range of numerical features of data by scaling using StandardScaler()[2]. Smaller weights are considered more regular or less specialized and as such, we refer to this penalty as weight regularization. Feature standardization makes the values of each feature in the data we have zero-mean (when subtracting the mean in the numerator) and unit-variance. In scaling, we have used standard scalar from ski-kit learn to scale numerical features except for *idnum, wages and wageclass*.



To check correlations between numerical variables, derived from sns()[3] heatmap.

Correlation Matrix Insights:

- As age increases, productivity and hours per week decreases.
- More hoursperweek related to more productivity and high wage.

One Hot Encoding:

We perform one-hot encoding to convert categorical features into a form that could be provided to ML algorithms to do a better job in prediction. After one-hot encoding 491 features were created for Data Modeling.

Model Selection using Cross-Validation and Parameter Tuning

Initially, we do predictions on whether a person is earning or not. As it is a classification problem, we will be using different classifiers for this case and choose the best one. Classification predictions will be better as compared to regression predictions as regression problems have real value while classification has binary value. We will take only predictions for a person classified as not earning in final predictions and predictions for the person who earns will be done using regressor. We also improved the classifiers and regressors by hyperparameter tuning with 5-fold Cross Validation using *GridSearchCV()*[4] and *RandomizedSearchCV()*[5]. We did 5 fold cross validation, as we have 1184 rows in the dataset, so we thought k = 5 will be an optimal value for it.

Model	Accuracy
Decision Tree Classifier	0.923941
Random Forest Classifier	0.972259

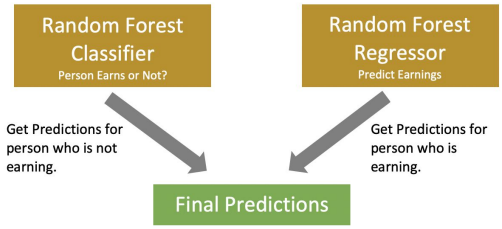
For binary classification, we used **Random Forest Classifier**[6] and **Decision Tree Classifier**[7] as these can efficiently classify large and complicated datasets and calculated the accuracy using Hyperparameter Tuning. Decision trees and Random Forest implicitly perform variable screening or feature selection. Therefore, we thought it is good for classification and chose this for binary predictions.

Based on cross validation score, **Random Forest Classifier** gives a good classification accuracy (as show below) and hence, we will use this for our final binary predictions.

The data given to us is non linearly separable data, and we don't have a certain way or equation to express the relationship between dependent and independent variables, so we decided to use **Decision Tree Regressor**[8] as it also allows us to do feature selection or variable screening, and it is not sensitive to outliers, **Random forest Regressor**[9] as it's simple and fast and **Support Vector Regressor**[10] **with linear kernel** because it provides flexibility in terms of distribution and avoids overfitting by allowing us to control cost parameters for penalizing the regression.

Model	Hyperparameters
Decision Tree Regressor	criterion = 'mse', max_depth = 2, max_leaf_nodes = 20, min_samples_leaf = 20, min_samples_split = 10
Random Forest Regressor	bootstrap = False, max_depth = 20, max_features = 'sqrt', min_samples_leaf = 1, min_samples_split = 10, n_estimators = 600
Support Vector Regressor	C=1000, gamma = 0.1, kernel = 'linear'

Model	Train Score	CV Score	RMSE
Decision Tree Regressor	0.4513	0.3814	46630.8111
Random Forest Classifier	0.8042	0.4146	33166.6664
Support Vector Regressor	0.2192	0.2594	36477.7967



Based on 5-fold Cross Validation, we can see that **Random Forest Regressor** is better at predicting. It also has low RMSE. Hence, we choose Random Forest Classifier to predict wag class and Random Forest Regressor for our final predictions and used it to predict wages in testing data.

Other things considered while performing Analysis

While trying to process data, we also tried binning the values for the feature *'workarrivaltime'* as there are too many values with not a very large difference between them. Also it would have helped in reducing the number of columns generated after performing one hot encoding, but when we did the binning, we didn't get expected results, we got almost zero information gain on '*workarrivaltime'* whereas initially, it was one of the top features as can be seen from the mutual information graph.

Work Citations

1. mutual_info_regression() -
https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.mutual_info_regression.html
2. StandardScaler() -
https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html
3. Seaborn sns() - https://seaborn.pydata.org/generated/seaborn.heatmap.html
4. GridSearchCV() -
https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html
5. RandomizedSearchCV() -
https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.RandomizedSearchCV.html
6. DecisionTreeClassifier() -
https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html
7. RandomForestClassifier() -
https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html
8. DecisionTreeRegressor() -
https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeRegressor.html
9. RandomForestRegressor() -
https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html
10. SVR() -
https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVR.html